*Dynamics of Continuous, Discrete and Impulsive Systems*
*Series* **B:** *Applications & Algorithms* **10 (2003)** 603-618


# Problem Investigations in TCP Flow Control Mechanisms

K.L. Eddie Law and Wing-Chung Hung

The Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto, Toronto, Ontario M5S 3G4

**Abstract.** Transmission Control Protocol (TCP) provides reliable process-to-process communication services between any two end systems across the Internet. With the TCP flow control mechanisms, multiple simultaneous TCP connections can share network resources. Since the late 1980s, there have been several popular TCP flow control designs. There include, for example, the Tahoe, Reno, Vegas and NewReno algorithms. Moreover, an TCP header option is designed to help the flow performance, especially in lossy conditions. The option field is known as Selective Acknowledgment (SACK). Though in this paper, the design problems of the popular flow control mechanisms are investigated, SACK is not examined. This is because SACK operates as an TCP option that explicitly marks down the missing blocks of information at the receivers. All other flow control families can deploy this option to improve their respective performance. On the other hand, many systematic flow control mechanisms were designed usually based on certain principles, network conditions and assumptions. With the advancements of technologies, the flow control designs suffer poor performance when many settings of the system parameters degrade their throughput performance rapidly if the networks are not in their favorable conditions. Through the problem investigations, we can improve the throughput performance by tackling each mistake accordingly in future.
**Keywords.** Transmission Control Protocol, design mistakes, Slow-Start, Congestion Avoidance, Fast Retransmit, Fast Recovery.

## 1 Introduction

The primary objective of the Transmission Control Protocol (TCP) [1] is to provide process-to-process reliable communication connections between any two computing hosts on the Internet. The TCP specifications [1] stipulate packet formats and finite state machines in order to ensure proper establishments and terminations of TCP connections. Thereafter, packet ordering between two communicating end systems can be monitored on per process basis. In order to offer reliability across all TCP connections, another recommendation [2], through the Internet Engineering Task Force (IETF), is